ELECTRICAL & COMPUTER ENGINEERING

# ECE 344L Course Summary

Spring 2020

## 1. Course Introduction

Computer hardware and software evolution

## 2. Number Formats

Bits
Bytes
Decimal
Binary
Hexadecimal
Words – vary by machine – MIPS – Word 32 bits
$X = \{x_{w-1} .. x_0\}$
Integer Formats
  Signed – value $= \sum_{i=0}^{w-1} x_i 2^i$
  Unsigned – value $= -x_{w-1}2^{w-1} + \sum_{i=0}^{w-2} x_i 2^i$
(Remember that bits are indexed as: $x_{w-1}..x_0$
Alignment requirements for storing integers in memory

## 3. Character Formats

ASCII
UTF-8

## 4. Computer Operations

Fetch – Decode – Execute

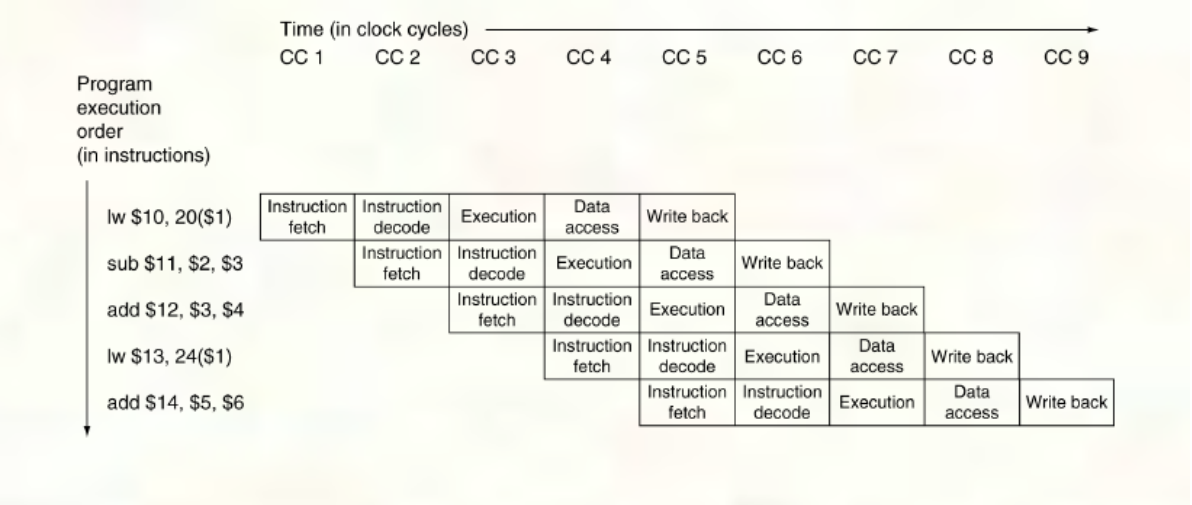
## 5. Computer Architecture

Harvard
Von Neumann
CISC
RISC
  Load/Store Architecture
  Fixed length instructions
  One instruction per clock cycle
  Limited addressing modes
  Many general-purpose registers – data and addresses

## 6. MIPS Processor

Virtual & Physical Memory addresses
Fixed Map Translation
Addressing: one mode – Base register + displacement
Stack operations
Register usage conventions
Assembly Language programming

## 7. MIPS processor pipeline




Data Hazards
Control Hazards

## 8. Microcontroller Characteristics – MIPS with integrated peripherals

Configuration using SFRs
Communication with peripherals with SFRs
Monitor peripheral status using SFRs

## 9. System Clocks

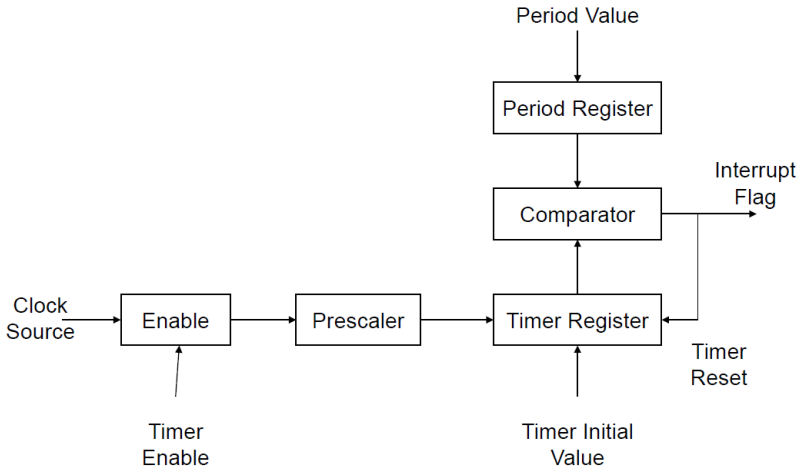
SYSCLK
PBCLK
USBCLK

***PERIPHERALS***

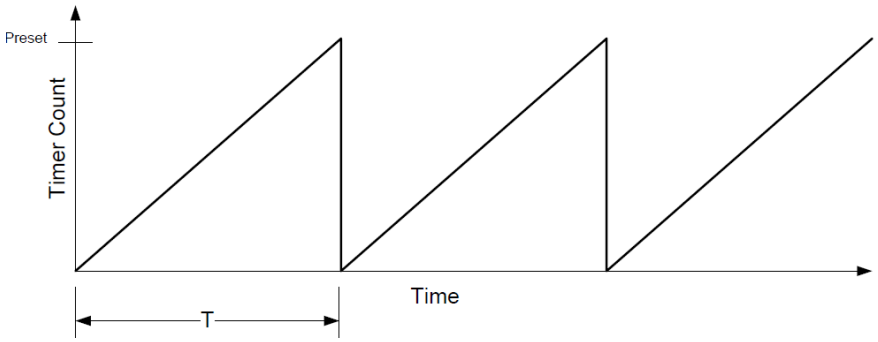
## 10. Ports

Digital I/O – inputs or outputs
Some shared with analog inputs
Open Drain configuration option

## 11. Timers




Type A & B timers
All are 16 bit or 32 bit with 2&3 or 4&5



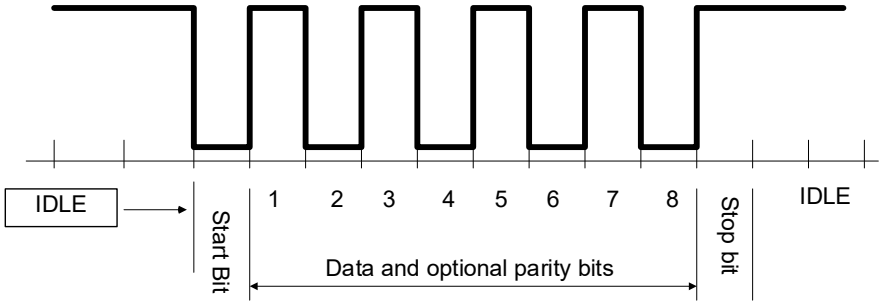

## 12. Communication Characteristics

Parallel – multiple bits simultaneously
Serial – One bit at a time
Synchronous – A clock is transmitted
Asynchronous – No transmitted clock
Full Duplex – Tx and Rx simultaneously
Half Duplex – Tx or Rx at any given time

## 13. Serial Communication Peripherals

**SPI** – synchronous, full duplex
**I2C, aka I²C or two-wire** – synchronous, half duplex

**UART** – asynchronous – fixed baud rate – extract synchronization info from signal




Character data encoding for transmission

ASCII

UTF-8

## 14. Interrupts

PIC32 uses an interrupt controller to provide support for up to 96 sources

Can be used in lieu of constant status polling for more efficient operation

### Requirements to use interrupts

- Need a source
- The specific interrupt for the source must be enabled
- Need an interrupt service routine to respond to the interrupt (ISR)

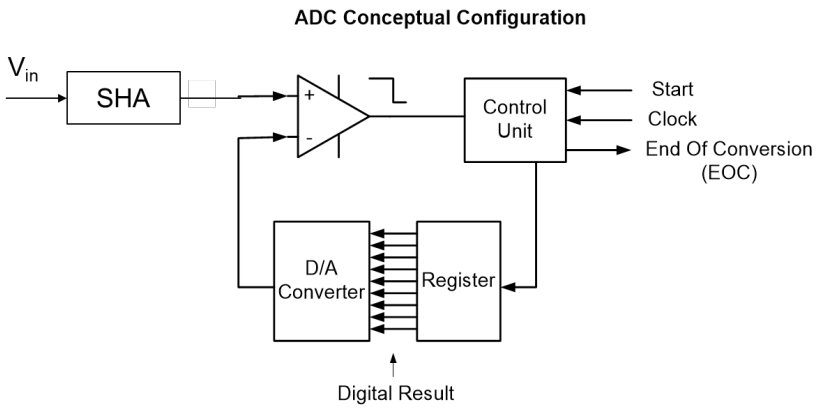
### ISR Requirements

- An ISR cannot be called directly by any other function
- Argument cannot be passed to an ISR
- An ISR cannot return any values
- Ideally, and ISR should not call any other functions

## 15. Processing Analog signals (Analog to Digital Converter – ADC)

Two types of ADC

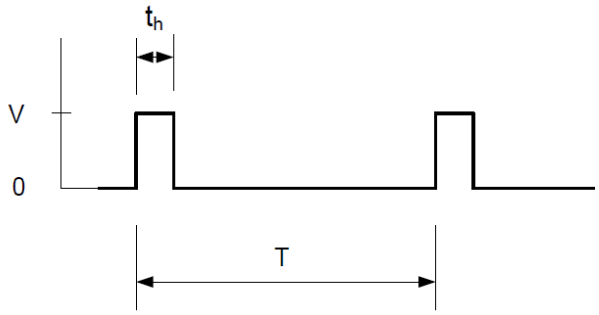
- Flash or direct conversion
- Successive Approximation

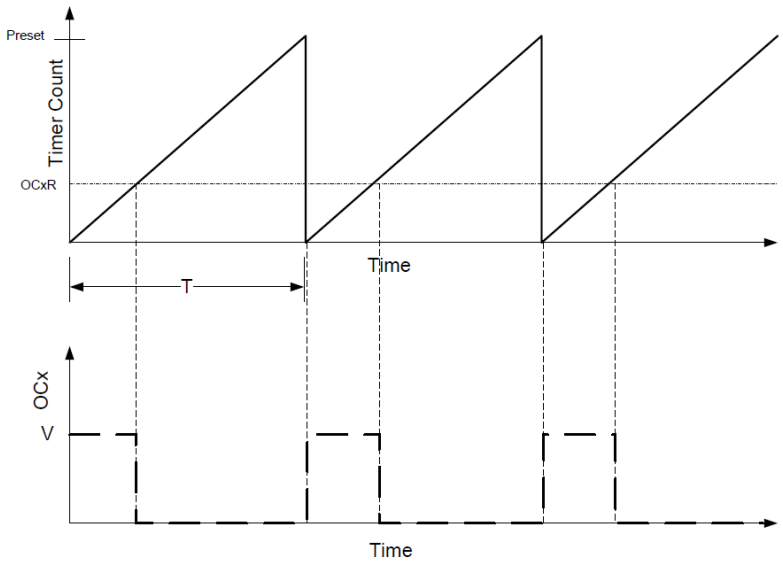



Sample processing time = acquisition + conversion time

## 16. **Pulse Width Modulated (PWM) Signal and Output Compare**

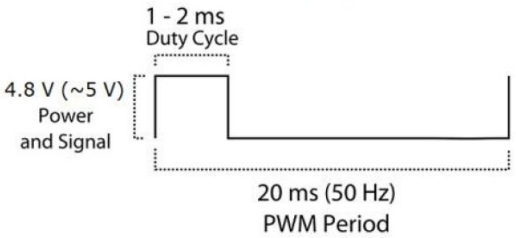
$$Duty\ Cycle = \left(\frac{t_h}{T}\right) * 100\ \%$$




## 17. **Servo Motors** – range of motion is limited

Position "0" (1.5 ms pulse) is middle, "90" (~2ms pulse) is middle,
is all the way to the right, "-90" (~1ms pulse) is all the way to the left.



18. **Stepper Motors** – full 360° range of motion – moves in step increments
    Multiple configuration – unipolar and bipolar windings
    Internal shaft rotation
        Full step – 11.25° /step
        Half step – 5.625° /step
    Output shaft rotation
        Full step – 0.1758° /step
        Half step – 0.0879° /step

    Moves in either direction by energizing phases one or two at a time.

(Unipolar)

**One Phase Half Step Sequencing**

| Step | A | C | B | D |
|---|---|---|---|---|
| **1** | **1** | 0 | 0 | 0 |
| **2** | **1** | **1** | 0 | 0 |
| **3** | 0 | **1** | 0 | 0 |
| **4** | 0 | **1** | **1** | 0 |
| **5** | 0 | 0 | **1** | 0 |
| **6** | 0 | 0 | **1** | **1** |
| **7** | 0 | 0 | 0 | **1** |
| **8** | **1** | 0 | 0 | **1** |